Quantum Computers.
How to Program Them?

Hakob Avetisyan
Gate42
www.gate42.org
https://github.com/gate42qc

Topics

Part 1 (~40 min):

What is a quantum computer?

How are quantum computers different than conventional computers?

Universal Quantum Algorithms

Quantum Software

Part 2 (~1 hour):

Linear Algebra

Programming in Qiskit,

Teleportation Demo on IBM hardware and Simulator using Qiskit

What is a quantum computer?

QC $\neq$ smaller, faster CC

Fundamentally new paradigm for processing information

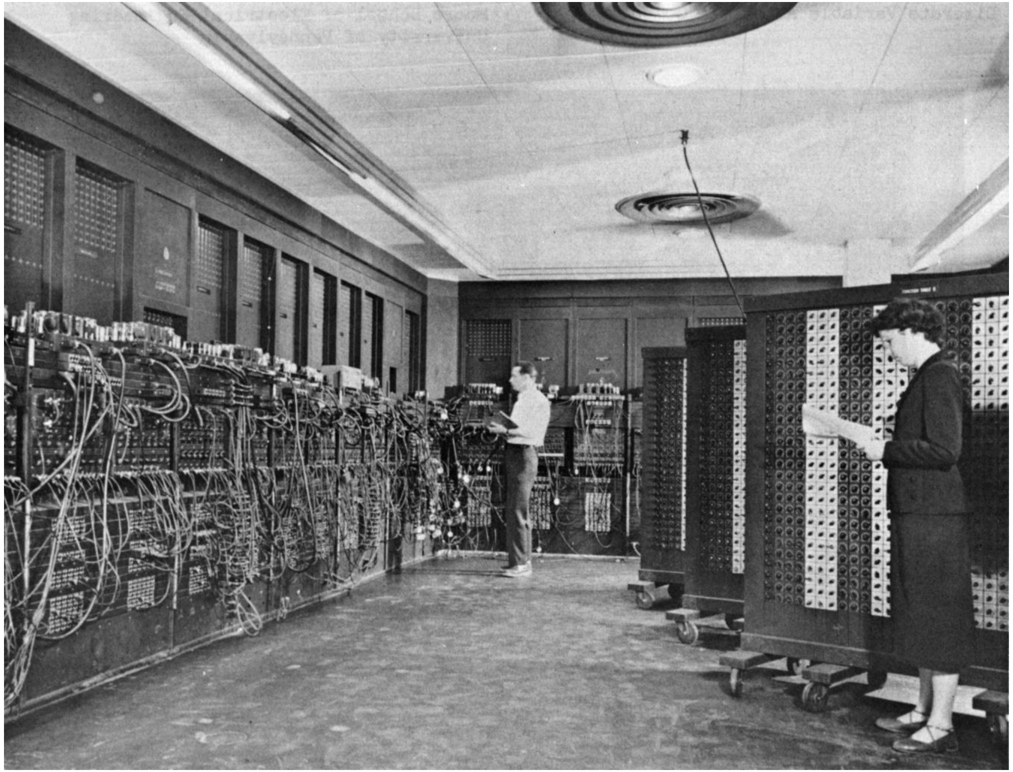
Violate strong Church-Turing thesis

Potential to exceed the performance of CC for problems such as:

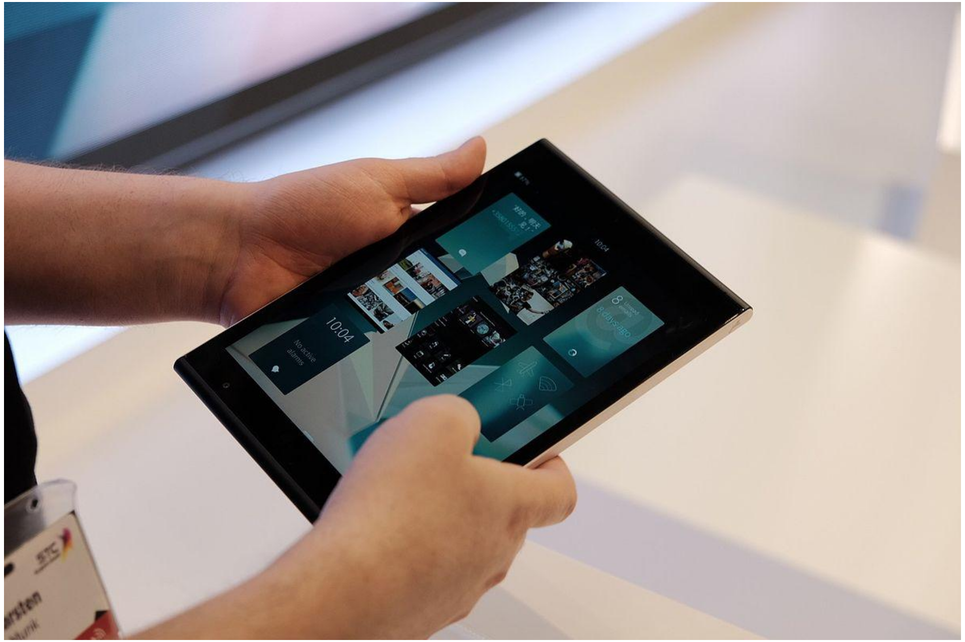
1. Cybersecurity
2. Materials science
3. Chemistry
4. Pharmaceuticals
5. Machine learning
6. Optimization

Over past 80 years CC's have changed dramatically

from the room-filling vacuum-tube-based computers like ENIAC...



....to the tablets and phones people use today.



What does a QC look like?

Currently, QCs look like an ENIAC than like a laptop or tablet

How Is a Quantum Computer Different?

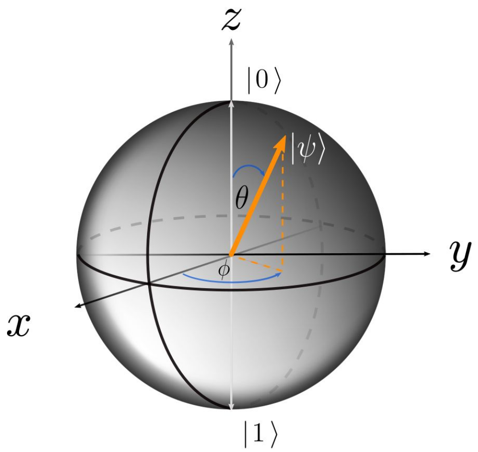
- For N bits, there are 2^N possible classical states.
- CC can represent only one of these N-bit states at a time.
- Qubits in a quantum computer can be set into a single superposition state that may simultaneously carry aspects of all 2^N components

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$$

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|\psi\rangle = \begin{bmatrix} \cos(\theta/2) \\ e^{i\phi}\sin(\theta/2) \end{bmatrix}$$

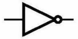
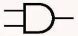
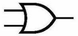
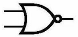
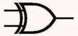
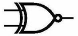


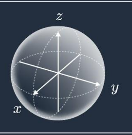
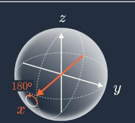
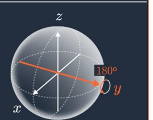
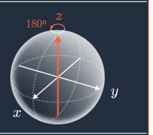
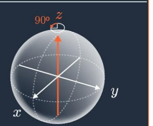
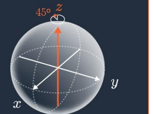
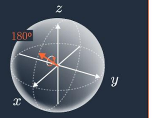
	Bits	Probabilistic Bits	Qubits
State (Single Unit)	Bit $\in \{0, 1\}$	Real Vector $\vec{s} = a\vec{0} + b\vec{1}$ $a, b \in R_+$ $a + b = 1$	Complex Vector $\lvert\psi\rangle = \alpha\lvert 0\rangle + \beta\lvert 1\rangle$ $\alpha, \beta \in C$ $\lvert\alpha\rvert^2 + \lvert\beta\rvert^2 = 1$
State (Multi Unit)	Bitstring $\in \{0, 1\}^n$	Prob. Distribution (Stochastic Vector) $\vec{s} = \{p_x\}_{x \in \{0,1\}^n}$	Wavefunction (Complex Vector) $\lvert\psi\rangle = \{\alpha_x\}_{x \in \{0,1\}^n}$
Operations	Boolean Logic	Stochastic Matrices $\sum_{j=1}^{S} P_{i,j} = 1$	Unitary Matrices $U^\dagger U = 1$
Component Ops	Boolean Gates	Tensor Product of Matrices	Tensor Product of Matrices

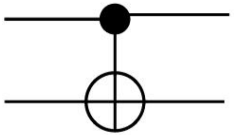
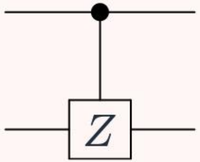
$$|\psi\rangle = \alpha_0 |00\cdots 0\rangle + \alpha_1 |00\cdots 01\rangle + \cdots \alpha_{2^{n-1}} |11\cdots 1\rangle$$

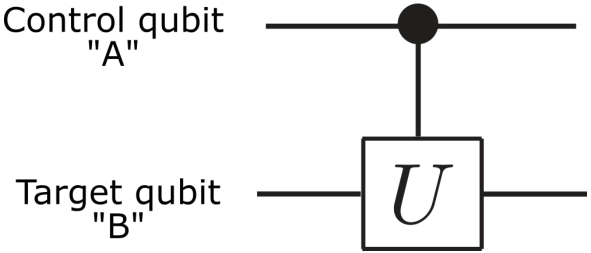
$$|\phi\rangle = a|00\rangle - b|11\rangle \longleftarrow$$ **Entangled State**

Classical and Quantum Gates

GATE		CIRCUIT REPRESENTATION	TRUTH TABLE Input	Output
NOT	The output is 1 when the input is 0 and 0 when the input is 1.		0 1	1 0
AND	The output is 1 only when both inputs are 1, otherwise the output is 0.		0 0 0 1 1 0 1 1	0 0 0 1
OR	The output is 0 only when both inputs are 0, otherwise the output is 1.		0 0 0 1 1 0 1 1	0 1 1 1
NAND	The output is 0 only when both inputs are 1, otherwise the output is 1.		0 0 0 1 1 0 1 1	1 1 1 0
NOR	The output is 1 only when both inputs are 0, otherwise the output is 0.		0 0 0 1 1 0 1 1	1 0 0 0
XOR	The output is 1 only when the two inputs have different value, otherwise the output is 0.		0 0 0 1 1 0 1 1	0 1 1 0
XNOR	The output is 1 only when the two inputs have the same value, otherwise the output is 0.		0 0 0 1 1 0 1 1	1 0 0 1

GATE	CIRCUIT REPRESENTATION	MATRIX REPRESENTATION	TRUTH TABLE Input	Output	BLOCH SPHERE
I Identity-gate: no rotation is performed.	$\boxed{I}$	$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\|0\rangle$ $\|1\rangle$	$\|0\rangle$ $\|1\rangle$	
X gate: rotates the qubit state by π radians ($180°$) about the x-axis.	$\boxed{X}$	$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\|0\rangle$ $\|1\rangle$	$\|1\rangle$ $\|0\rangle$	
Y gate: rotates the qubit state by π radians ($180°$) about the y-axis.	$\boxed{Y}$	$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$\|0\rangle$ $\|1\rangle$	$i\ \|1\rangle$ $-i\ \|0\rangle$	
Z gate: rotates the qubit state by π radians ($180°$) about the z-axis.	$\boxed{Z}$	$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\|0\rangle$ $\|1\rangle$	$\|0\rangle$ $-\|1\rangle$	
S gate: rotates the qubit state by $\frac{\pi}{2}$ radians ($90°$) about the z-axis.	$\boxed{S}$	$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$	$\|0\rangle$ $\|1\rangle$	$\|0\rangle$ $e^{i\frac{\pi}{2}}\|1\rangle$	
T gate: rotates the qubit state by $\frac{\pi}{4}$ radians ($45°$) about the z-axis.	$\boxed{T}$	$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$	$\|0\rangle$ $\|1\rangle$	$\|0\rangle$ $e^{i\frac{\pi}{4}}\|1\rangle$	
H gate: rotates the qubit state by π radians ($180°$) about an axis diagonal in the x-z plane. This is equivalent to an X-gate followed by a $\frac{\pi}{2}$ rotation about the y-axis.	$\boxed{H}$	$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\|0\rangle$ $\|1\rangle$	$\frac{\|0\rangle + \|1\rangle}{\sqrt{2}}$ $\frac{\|0\rangle - \|1\rangle}{\sqrt{2}}$	

GATE	CIRCUIT REPRESENTATION	MATRIX REPRESENTATION	TRUTH TABLE
Controlled-NOT gate: apply an X-gate to the target qubit if the control qubit is in state $\|1\rangle$		$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	Input / Output $\|00\rangle$ → $\|00\rangle$ $\|01\rangle$ → $\|01\rangle$ $\|10\rangle$ → $\|11\rangle$ $\|11\rangle$ → $\|10\rangle$
Controlled-phase gate: apply a Z-gate to the target qubit if the control qubit is in state $\|1\rangle$		$cZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$	Input / Output $\|00\rangle$ → $\|00\rangle$ $\|01\rangle$ → $\|01\rangle$ $\|10\rangle$ → $\|10\rangle$ $\|11\rangle$ → $-\|11\rangle$

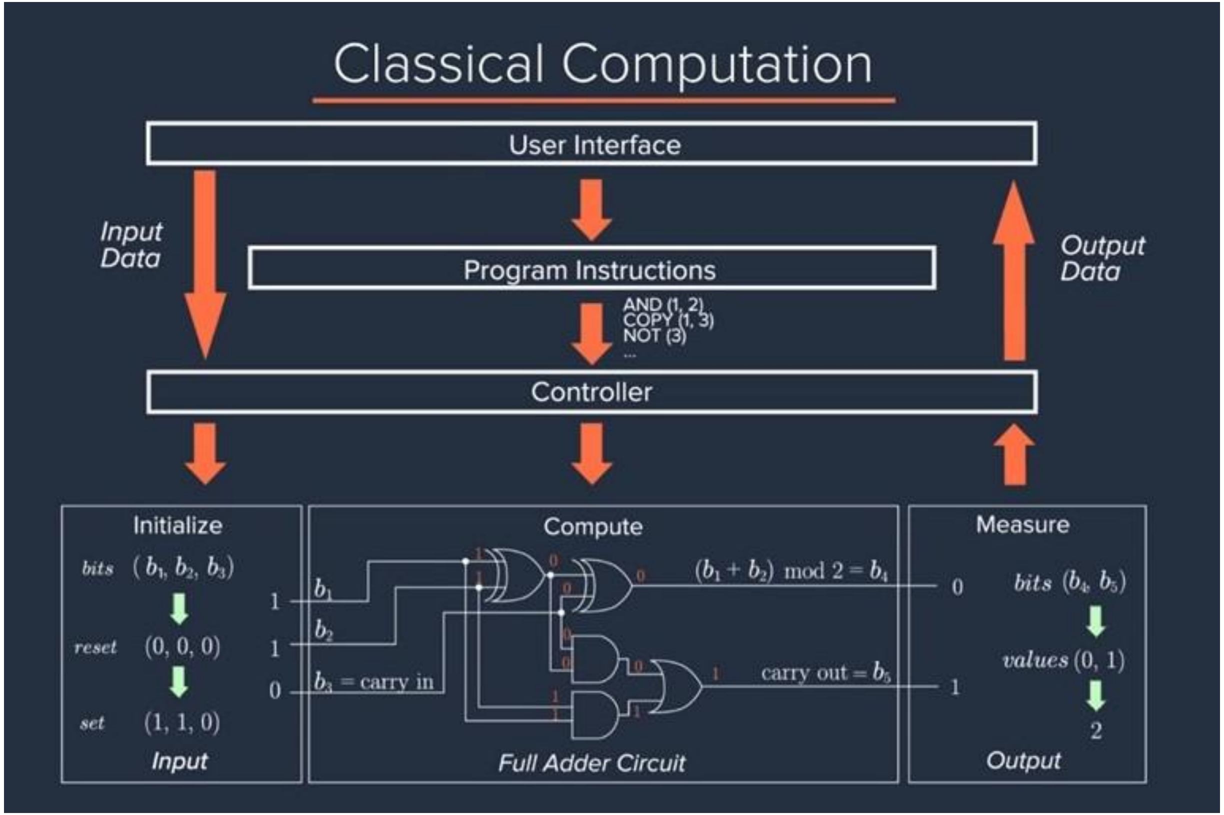


$$U^c_{AB} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix}$$

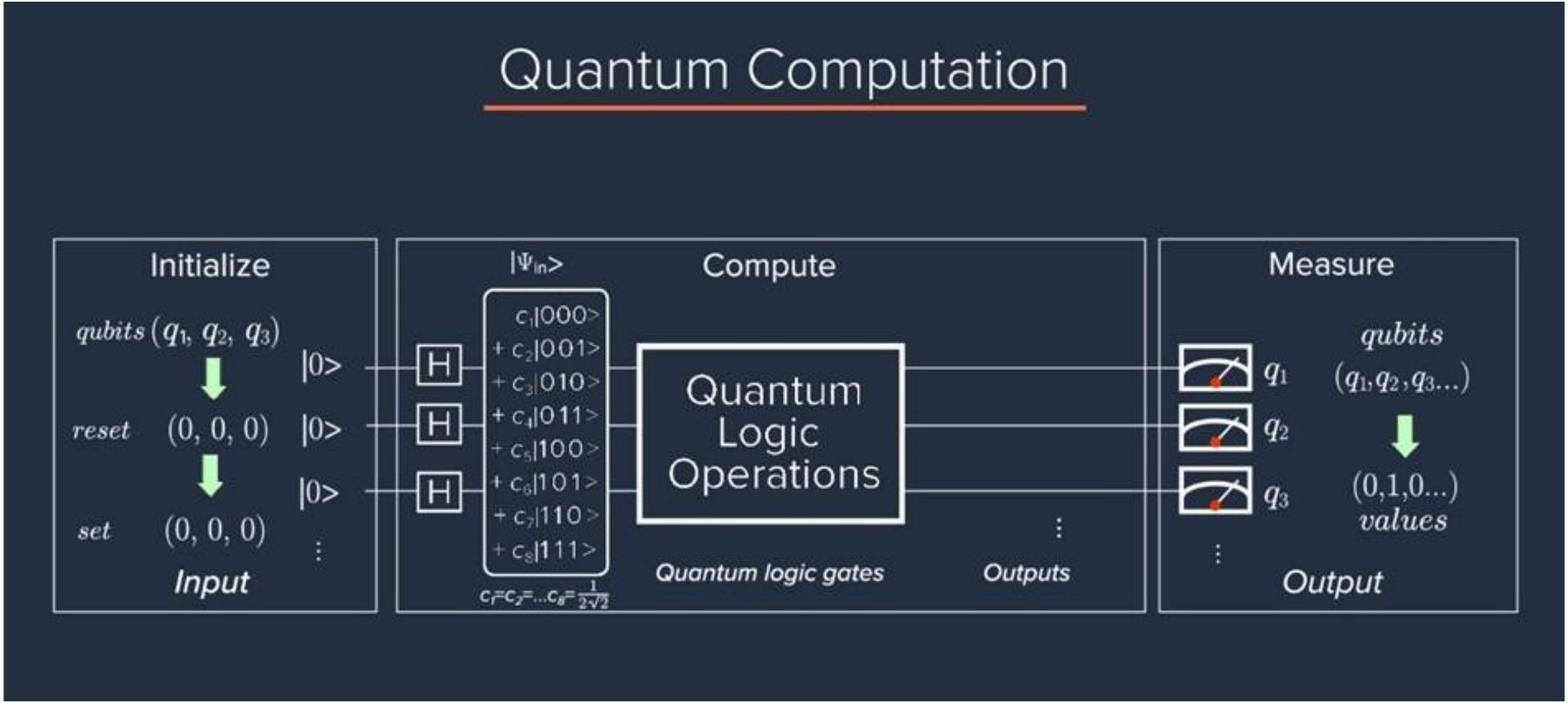
$$U^c_{A,B} = |0\rangle\langle 0|_A \otimes I_B + |1\rangle\langle 1|_A \otimes U_B$$

Universal Quantum Algorithms

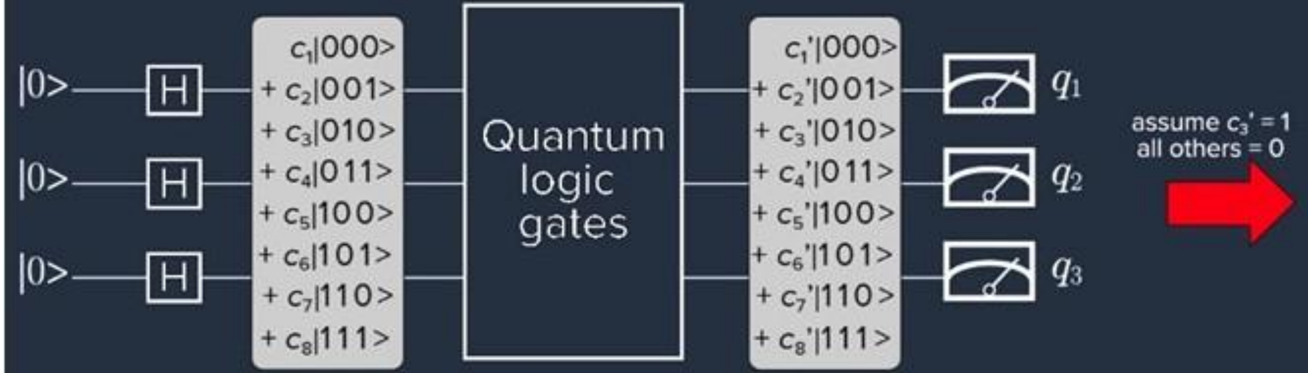
How a Universal Algorithm Works: Classical



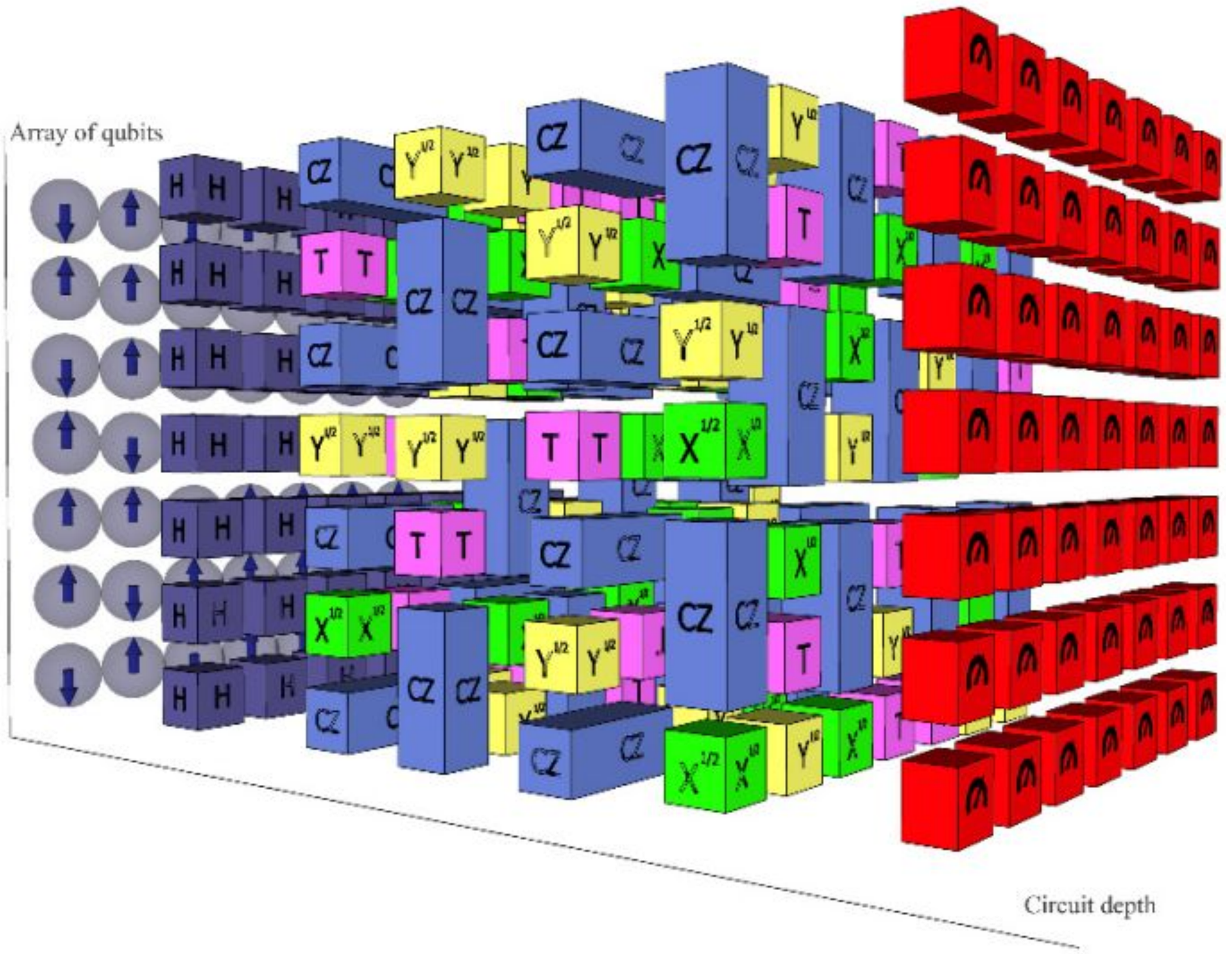
How a Universal Algorithm Works: Quantum



Measurement

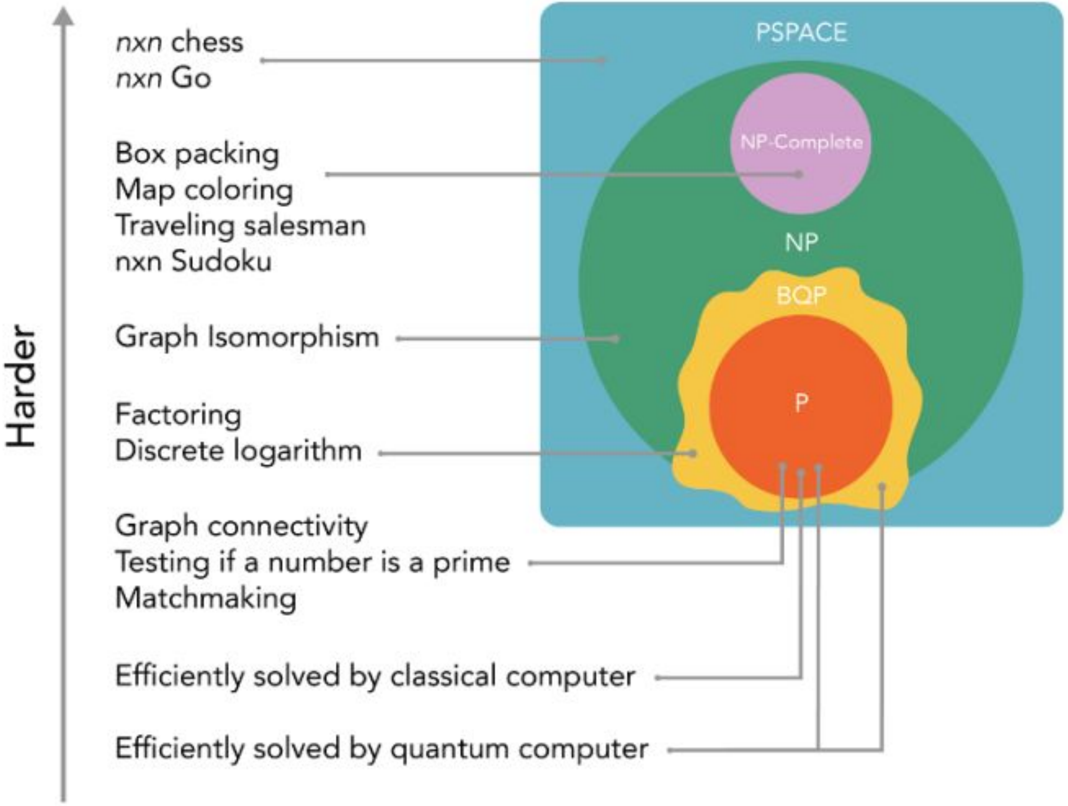


q_1	q_2	q_3	probability	
0	0	0	$\lvert c_1'\rvert^2$	0
0	0	1	$\lvert c_2'\rvert^2$	0
0	1	0	$\lvert c_3'\rvert^2$	1
0	1	1	$\lvert c_4'\rvert^2$	0
1	0	0	$\lvert c_5'\rvert^2$	0
1	0	1	$\lvert c_6'\rvert^2$	0
1	1	0	$\lvert c_7'\rvert^2$	0
1	1	1	$\lvert c_8'\rvert^2$	0


Array of qubits
Circuit depth

What Problems Can Be Solved by QC?

Example Problems



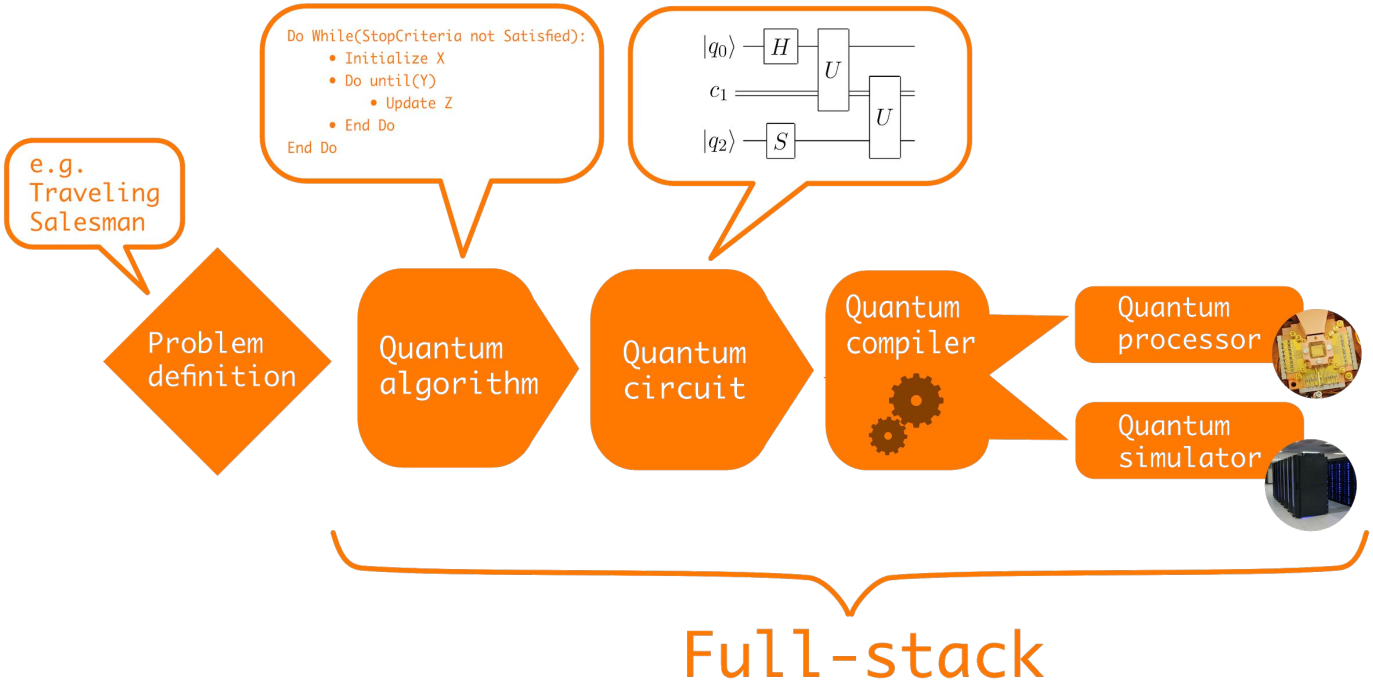
1. **Cybersecurity (NP)**
2. **Materials science (NP, PSPACE)**
3. **Chemistry (NP, PSPACE)**
4. **Nitrogen fixation (PSPACE)**
5. **Pharmaceuticals (PSPACE)**
6. **Machine learning (NP)**
7. **Optimization (NP)**

QUANTUM ALGORITHM SPEEDUPS

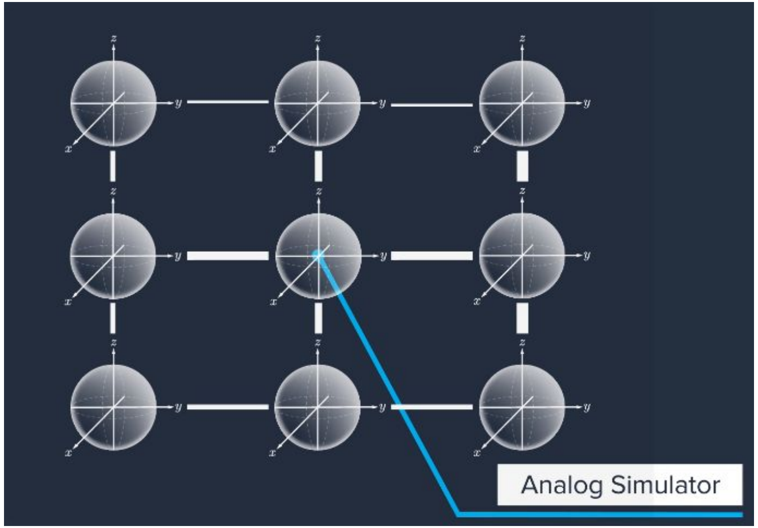
ALGORITHM	CLASSICAL RESOURCES	QUANTUM RESOURCES	QUANTUM ADVANTAGE	LIMITATION
Simulation (quantum chemistry)	2^N (for N atoms)	N^C	Exponential*	Mapping problem to qubits
Factoring (+ related number theoretic)	2^N (for N digits)	N^3	Exponential	Classical runtime limit unproven
Linear systems (Ax=b)	2^N (for N digits)	~N	Exponential	Strict conditions, e.g. sparse matrix
Optimization	2^N (for N items)	?	?	Empirical
Search (unsorted/unstructured data)	N (for N entries)	$\sqrt{N}$	Polynomial ($\sqrt{N}$)	Data loading

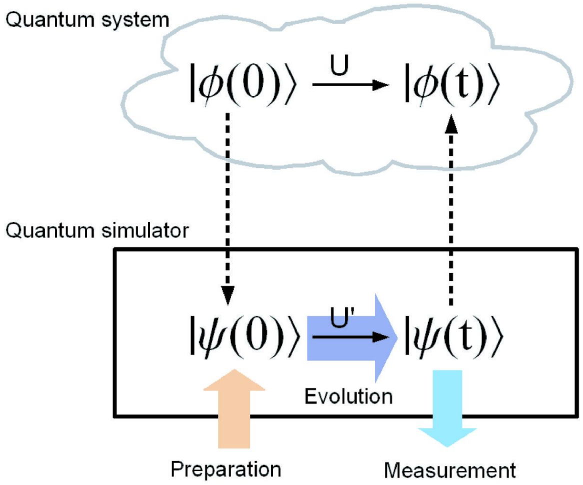
Types of Quantum Computers

Universal Gate Model Fault Tolerant

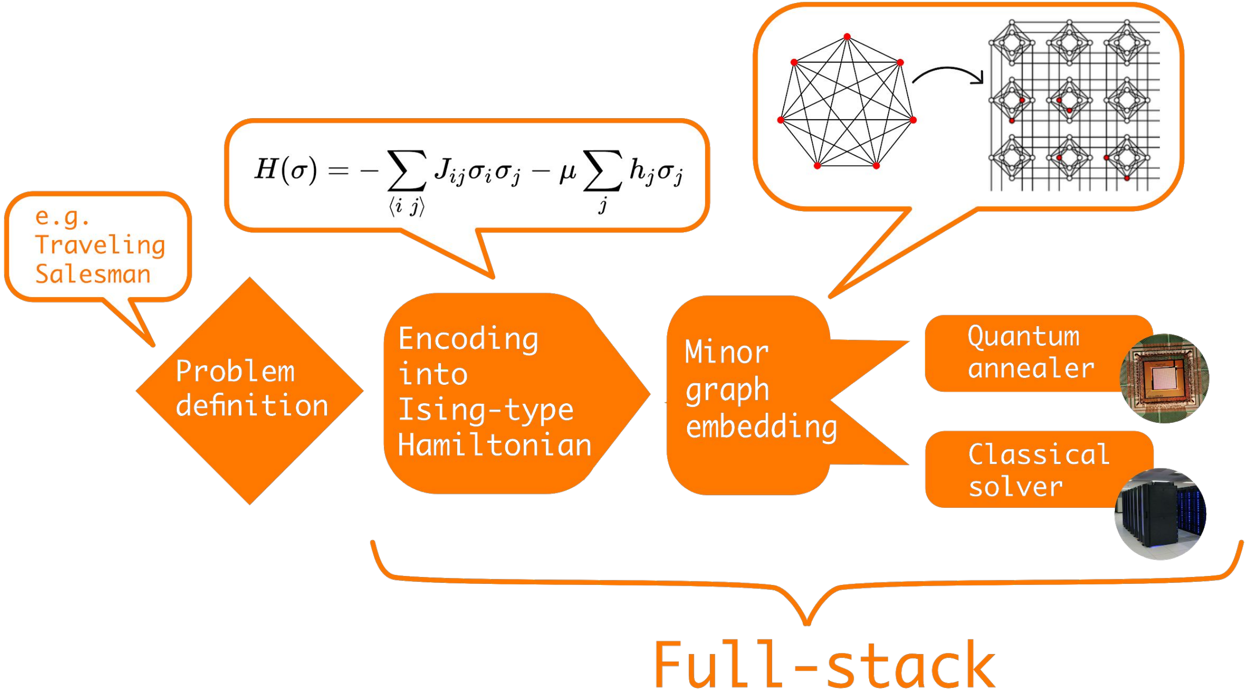


Quantum Simulator





Quantum Annealer

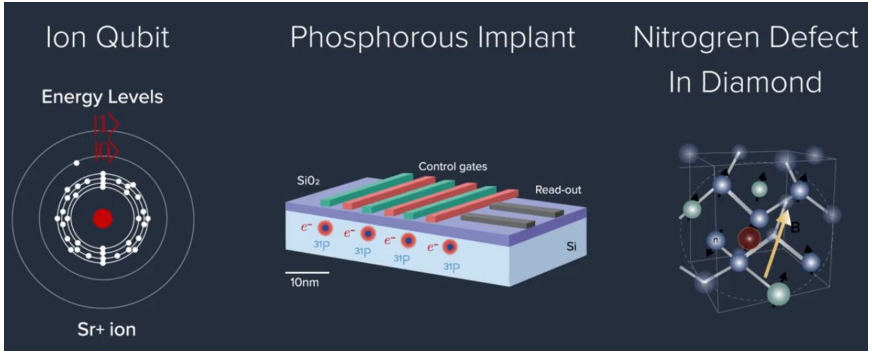


How to Build One?

Qubit Modalities and Di Vincenzo Criteria

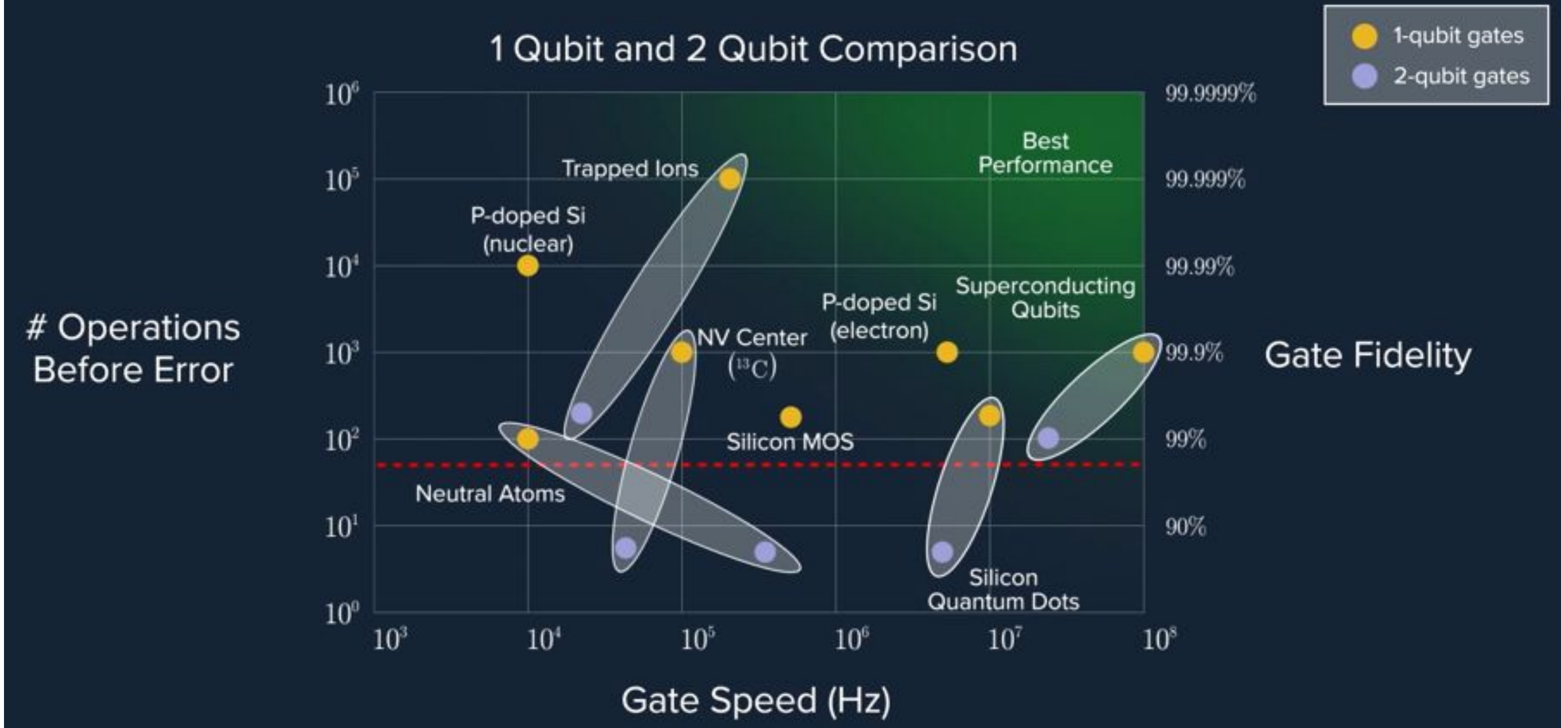
	D1 SCALABLE SYSTEM	D2 INITIALIZATION	D3 MEASUREMENT	D4 UNIVERSAL GATES	D5 COHERENCE	D6 INTERCONVERSION	D7 COMMUNICATION
SiGe Quantum Dots	Yellow	Green	Green	Green/Yellow	Green	Yellow	Yellow
Doped Silicon	Yellow	Green/Yellow	Green	Yellow	Green	Yellow	Yellow
NV Centers	Yellow	Yellow	Green	Green/Yellow	Green	Green	Green
Neutral Atoms	Green/Yellow	Green	Yellow	Yellow	Green	Green	Green
Trapped Ions	Green/Yellow	Green	Green	Green	Green	Green/Yellow	Green
Superconductors	Green/Yellow	Green	Green	Green	Green/Yellow	Green/Yellow	Green

- Green: Sufficient demonstrations exist to proceed to the 100 qubit level
- Yellow: Concepts and/or first demonstrations exist



		SUPER-CONDUCTING	TRAPPED IONS	SILICON QUANTUM DOTS		IMPURITY-DOPED SILICON		IMPURITY CENTERS	TRAPPED NEUTRAL IONS
QUBIT MODALITY	MATERIALS	Al	Yb+, Ca+, Sr+, Be+, Ba+, Mg+	isotopically purified Si, SiGe	isotopically purified Si	Phosphorus doped isotopically purified silicon		Nitrogen vacancies in diamond	Rb, Cs, Ho
	TYPE	transmom	hyperfine & optical transitions	quantum dot 2DEG	quantum dot Si MOS	implanted dopant		nuclear + electron spin	Rydberg states
	CONTROL/ RO	microwaves	microwaves & optics	baseband or microwave	baseband or microwave	microwaves		microwaves & optics	uwave
	STATE	junction phase	atomic state of electron	electron spin	electron spin	electron spin	nuclear spin	nuclear + electron spin	Rydberg state
STATE-OF-ART TIMES (ns)	T1	50,000	>1E14 (years)	1,000,000,000	2,000,000,000	5,000,000,000	>1E14	100,000,000	5,000,000,000
	T2	100,000	50,000,000,000	400,000	1,200,000	100,000	1,000,000,000	20,000,000	1,000,000,000
	1QB GATE	10	5,000	100	2,000	200	100,000	10,000	100,000
	2QB GATE	40	50,000	200	1,000	TBD	TBD	25,000	3,000
	RO	200	30,000	1,000	100,000	1,000,000	50,000,000	50,000	10,000,000
FIDELITY	1QB	99.90%	99.999%	99.60%	99.5%	99.90%	99.99%	99.90%	99%
	2QB	99.0%	99.5%	80%	TBD	TBD	TBD	82%	80%
	RO	99%	99.99%	99%	95%	95.00%	99.90%	94%	99.90%
CLOCK (MHz)	1QB GATE	100.00	0.20	10.00	0.50	5.00	0.01	0.10	0.01
	2QB GATE	25.00	0.02	5.00	1.00	TBD	TBD	0.04	0.33
	READOUT	5.00	0.03	1.0E+00	1.0E-02	1.0E-03	2.0E-05	2.00E-02	1.00E-04

1 Qubit and 2 Qubit Fidelity and Gate Speed



Quantum Software

- How do we efficiently design a quantum processor?
- How do we validate that it is working properly?
- How we do program it?

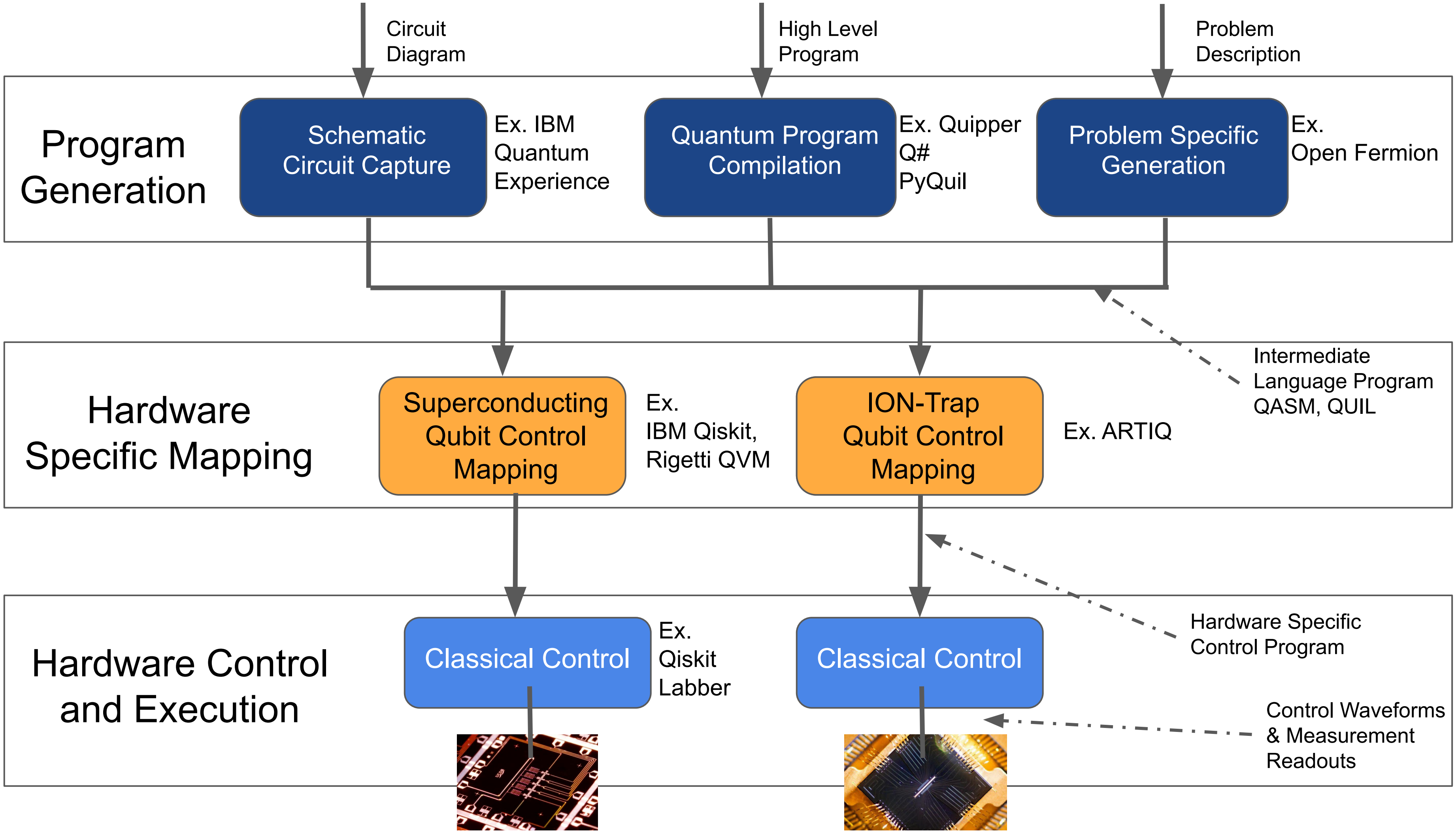
The 3 primary types of software

1. Program generation software (technology agnostic)

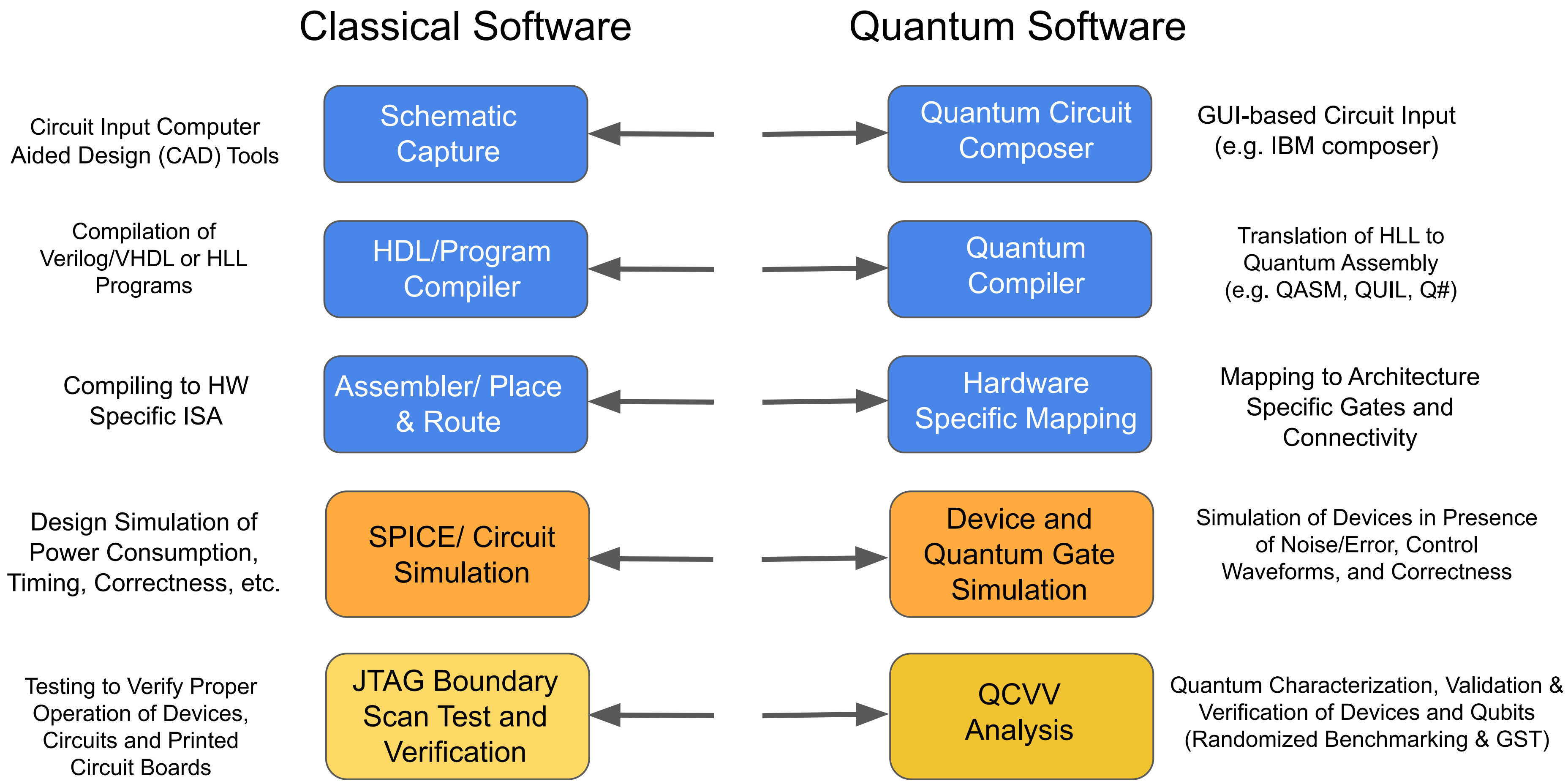
- Schematic capture (generally related to the quantum circuit model)
- HLL: high-level programming languages -- (C, Fortran, Pascal - abstracting away subroutines)
- Lower-level assembly languages (abstracting away specific hardware controls and subroutines)
- Problem-specific platforms

2. Circuit mapping software (require knowledge of the hardware architecture)

3. Cntrl and execution software (require knowledge on hardware architecture)



Circuit Diagram
High Level Program
Problem Description
Program Generation
Schematic Circuit Capture
Ex. IBM Quantum Experience
Quantum Program Compilation
Ex. Quipper Q# PyQuil
Problem Specific Generation
Ex. Open Fermion
Intermediate Language Program QASM, QUIL
Hardware Specific Mapping
Superconducting Qubit Control Mapping
Ex. IBM Qiskit, Rigetti QVM
ION-Trap Qubit Control Mapping
Ex. ARTIQ
Hardware Specific Control Program
Hardware Control and Execution
Classical Control
Ex. Qiskit Labber
Classical Control
Control Waveforms & Measurement Readouts


Classical Software
Quantum Software
Circuit Input Computer Aided Design (CAD) Tools
Schematic Capture
Quantum Circuit Composer
GUI-based Circuit Input (e.g. IBM composer)
Compilation of Verilog/VHDL or HLL Programs
HDL/Program Compiler
Quantum Compiler
Translation of HLL to Quantum Assembly (e.g. QASM, QUIL, Q#)
Compiling to HW Specific ISA
Assembler/ Place & Route
Hardware Specific Mapping
Mapping to Architecture Specific Gates and Connectivity
Design Simulation of Power Consumption, Timing, Correctness, etc.
SPICE/ Circuit Simulation
Device and Quantum Gate Simulation
Simulation of Devices in Presence of Noise/Error, Control Waveforms, and Correctness
Testing to Verify Proper Operation of Devices, Circuits and Printed Circuit Boards
JTAG Boundary Scan Test and Verification
QCVV Analysis
Quantum Characterization, Validation & Verification of Devices and Qubits (Randomized Benchmarking & GST)

Open Source Development Libraries for QC Programming

1. **Build a quantum circuit consisting of quantum and/or classical registers**
2. **Add operations to the circuit**
3. **Simulate the circuit**

Company	Name	Links	OS	Classical Language	Quantum Language
IBM	Qiskit	https://qiskit.org/	Mac, Windows, Linux	Python	OpenQASM
Rigetti	Forest	https://www.rigetti.com/forest	Mac, Windows, Linux	Python	QUIL
Google	CirQ	https://github.com/quantumlib/Cirq	Mac, Windows, Linux	Python	
Xanadu	StrawberryFields	https://strawberryfields.readthedocs.io/en/latest/	Mac, Windows, Linux	Python	Blackbird
D-Wave	Ocean SDK	https://www.dwavesys.com/take-leap	Mac, Windows, Linux	Python	QMASM
Microsoft	QDK	https://www.microsoft.com/en-us/quantum/development-kit	Mac, Windows, Linux	Python	Q#

All available software platforms at https://quantiki.org/wiki/list-qc-simulators